

## GUEST EDITORIAL PREFACE

# Research Challenges in Information Science (RCIS'2013)

*Roel Wieringa, Information Systems Group, University of Twente, Enschede, Netherlands*

*Selmin Nurcan, Center for Computer Research, University Paris 1 Panthéon-Sorbonne, Paris, France*

The series of the IEEE International Conferences on Research Challenges in Information Science (RCIS) aims at providing an international forum for scientists, researchers, engineers and developers on a wide range of information science areas to exchange ideas and approaches in this evolving field. In this special issue, we present five papers that are based on six of the best papers presented at the seventh IEEE International RCIS conference, held in 2013 in Paris. The six best papers of the conference (among 45 full papers selected over 175 submissions) have been asked to submit extended version to a new, and independent review process for IJSDM. Five of these were conditionally accepted and, after incorporation of final changes, included in this special issue.

The RCIS conference stimulates exchange of ideas on a wide variety of topics. The five papers in this issue treat topics in model-driven engineering, intention mining, business intelligence, requirements engineering, and information search & discovery.

Ormeño et al. describe an extension to a state-of-the-art model-driven development

method with techniques to generate usability requirements. The authors have collected user interface design guidelines and organized these in a tree-structured dialog that supports the analyst when eliciting usability requirements from the end user. This has been implemented as an extension to the OO\_method MDD tool and tested in a laboratory demonstration. The paper is one more step in the direction of full model-driven development, where all system features are included in a system model from the start of the development process.

Deneckère et al. take a significant step on the way to the ambitious goal to mine people's intentions from a log of their activities. The space of possible intentions must be known in advance, and it is assumed that actors work towards a common goal. This is the case in for example in business processes where strategic objectives and compliance goals need to be achieved. Using hidden Markov model techniques, the authors are able to high levels of recall and precision. The approach has been tested in two laboratory experiments.

Lapouchnian et al. explore the boundaries of the i\* requirements modeling language in the face of the adaptive enterprise. They model a hypothetical business intelligence application that evolves from a traditional application with stable requirements specified up-front, into an analytical toolbox that adapts itself to the needs of power users. Along the way, they show that the i\* notation is able to express actors, objectives, assumptions and dependencies in a sociotechnical system, but that it cannot express solution dynamics or express cyclical patterns of handling change. They identify several modeling challenges and issues that must be resolved in a requirements language for adaptive enterprise architecture.

Macasaet et al. have developed tools and techniques for identifying requirements patterns for small businesses and for linking these patterns to software components that can realize them. They identified a library of requirements patterns for inventory systems. They have developed a very pragmatic approach and tested

it in practice. They also have made their results available on the web and provide guidelines for others to test and use this in practice. This approach allows small business owners to relate to the requirements specification and allows software developers to reuse software components to rapidly implement a system that meets the requirements.

Lecarpentier et al. present a model-driven web engineering framework called Sydonie, that can manage documents expressed in different languages and embodied in different manifestations, including data and metadata at the same level. The approach is flexible and can be used in an agile development process. The paper presents the conceptual model of the approach, the implementation in a prototype, and an evaluation of the prototype. The approach is valuable when applications need to be created that do not fit existing software systems, agility is important, and management of metadata is important.

*Selmin Nurcan is Director of the Master Information and Knowledge Systems at the University Paris 1 Panthéon-Sorbonne and a senior researcher at the 'Centre de Recherche en Informatique' (CRI). Her research activities include enterprise computing, business process management, change modeling, business/IS alignment, IS governance, process (re)engineering and IS engineering and CSCW. She is co-organiser of the BPMDS series at CAISE, the BPMS2 workshop series at BPM since 2008, the SoEA4EE workshop series at EDOC since 2009. She is acting as a program committee member of a number of international conferences and workshops and she is serving on the editorial board of International Journal of Information System Modeling and Design, Requirements Engineering Journal, International Journal of Information Systems in the Service Sector, and the e-journal on Advances in Enterprise Systems.*

*Roel Wieringa (<http://www.cs.utwente.nl/~roelw>) is Chair of Information Systems at the University of Twente, The Netherlands. He has done research in many aspects of requirements engineering and information systems design, ranging from the application of formal methods and modal logics in the information systems specification to the development of methods for designing IT coordination support in business networks. His current interests are requirements and design of dynamic service-oriented networks, risk assessment for IT security and safety, and design research methodology. He has written two books, on Requirements Engineering and on the Design of Reactive Systems. His next book, Design Science Methodology for Information Systems and Software Engineering will appear in 2014 with Springer.*